基于循环神经网络和注意力模型的文本情感分析 *

胡荣磊，芮　璐，齐　筱，张昕然

(北京电子科技学院 电子与通信工程系，北京 100070)

摘　要：文本情感分析是自然语言处理领域的一大分支，其发展不仅在自然语言处理领域有重大的影响，更是在政治、经济、社会科学等受人们主观观点影响较大的领域有着深远的影响。深度学习在文本情感分析中的作用越来越重要。将循环神经网络中的长短期记忆网络和前馈注意力模型相结合，提出一种文本情感分析方案。在基本长短期记忆网络中加入前馈注意力模型，并在 Tensorflow 深度学习框架下对方案进行了实现。根据准确率、召回率和 F1 测度等衡量指标，与现有的方案对比表明，提出的方案较传统的机器学习方法和单纯的长短期记忆网络的方法有明显的优势。

关键词：文本情感分析；深度学习；长短期记忆模型；注意力模型

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2018.05.0300

Text sentiment analysis based on recurrent neural networks and attention model

Hu Ronglei, Rui Lu, Qi Xiao, Zhang Xinran

(*Department of electronic & Communication Engineering of Beijing Electronic Science &Technology Institute*, *Beijing* 100070, *China*)

Abstract: Sentiment analysis is a big branch in the field of natural language processing. The development of text sentiment analysis not only has a great influence in the field of natural language processing, but also has a profound influence in the fields of politics, economy and social sciences which are influenced by people's subjective view. The role of deep learning in text sentiment analysis is becoming more and more important. A text sentiment analysis scheme was proposed based on Long Short-Term Memory(LSTM) networks and feedforward attention model. It adopts feedforward attention model in basic Long Short-Term Memory networks. The scheme was implemented under the framework of Tensorflow depth learning. According to the parameters of precision, recall and F1 measures, the scheme was compared with the existing scheme. The results show that the model proposed in this paper has improved significantly compared with the traditional machine learning method and the traditional Long Short-Term Memory networks.

Key words: sentiment analysis; deep learning; long short-term memory; attention model

0　引言

近年来，互联网信息的爆炸式增长使得自然语言处理占据着至关重要的位置，无论是学术界还是工业界，对海量文本及数据的处理都颇为感兴趣。文本情感分析作为自然语言处理的一部分，其发展不仅在自然语言处理领域有重大的影响，更是在政治、经济、社会科学等受人们主观观点影响较大的领域有着深远的影响。

文本情感分析是指对包含用户表示的观点、喜好、情感等的主观性文本进行检测、分析以及挖掘。伴随着互联网的发展，人们在网上对产品、新闻、话题、时事事件等方面的内容都开始发表自己主观的观点及看法，分析大家的看法观点中的情感成为了中外学者近年来的研究热点。

随着互联网的发展，文本数据规模的不断增加，利用深度学习进行文本情感分析逐渐发展起来。由于深度学习在各个领域的成功应用和在文本情感分析领域的应用，本文提出一种基于长短期记忆网络和注意力模型的结构来进行文本情感分析的工作，并进行试验，经过验证，本文提出的方法是有效的。

1　相关工作

文本情感分析常用的方法主要是基于情感词典的文本情感分析和基于传统机器学习算法的文本情感分析。基于词典的方法主要是通过匹配情感词典中的情感词，通过一定的规则，对文本进行评分，最终得出情感极性的过程。Kim 和 Hovy 提出

收稿日期：2018-05-16；**修回日期：**2018-07-16　**基金项目：**国家自然科学基金面上项目（61772047）；中央高校基本科研业务费资助项目（2017LG-01）

作者简介：胡荣磊（1977-），男，河北衡水人，副研究员，博士，主要研究方向为无线通信与信息安全（huronglei@sina.com）；芮璐（1993-），女，硕士研究生，主要研究方向为情感分析；齐筱（1993-），女，硕士研究生，主要研究方向为无线网络安全、侧信道分析；张昕然（1987-），男，黑龙江哈尔滨人，讲师，博士，主要研究方向为无线信号处理与网络.

一种基于词典的算法[1]，该方法以 WordNet 情感词典中的近义词反义词关系以及一些给定的带有积极和消极情感的情感词为基础，采用引导策略构成新的情感词典，通过乘以句子中情绪词的分数来确定句子的情感方向。基于词典的文本情感分析方法在一段时期内成为主流，并且诞生了各种词典，具体可参见文献[2]。但由于需要过多的人工标注数据，该方法费时费力。目前基于词典的分析方法已经用的比较少了。

基于常规的机器学习算法的方法主要是将文本情感分析问题看作是文本分类问题来解决的。由于情感分类是将文本分为积极和消极，所以情感分析更加注重对表达情感的词语和句子的研究。Pang 等人[3]最先发表论文提出了采用监督学习（supervised learning）算法进行文本情感分类，他们分别使用朴素贝叶斯（naive Bayes），最大熵（maximum entropy）和支持向量机（support vector machine，SVM）等监督学习方法对影评文本进行了情感分类。大量研究表明，支持向量机算法在文本情感分析方面效果优秀。这些方法也有其缺点，实现虽然容易，但泛化能力较差，在某一领域表现优秀的算法换个领域可能将表现很差。

近年来，随着深度学习在自然语言处理方面的应用，自然语言处理有了很大的发展。而文本情感分析也越来越多地用到了深度学习算法。Bengio[4]最早使用神经网络来构建语言模型，由于该模型参数很多，训练成本很高，所以谷歌公司的 Mikolov 等人[5]通过简化该模型，提出了 word2vec 模型，实现了 CBOW 和 Skip-gram 两种框架且被用于多个领域。Kim[6]提出使用卷积神经网络（convolutional neural network，CNN）进行句子建模，解决情感分类任务，在多个数据集上均取得了很好的结果。Santos 等[7]利用深度卷积神经网络对短文本进行了情感分析。Irsoy 等人[8]使用循环神经网络（recurrent neural networks，RNN）为句子建模，而长短期记忆模型（long short-term memory，LSTM）作为循环神经网络的一个模型，也被 Tai 等人[9]证实能解决情感分析问题。LSTM 是一种特殊的循环神经网络，在许多任务中，LSTM 模型表现得比标准的循环神经网络要出色得多。注意力模型由 Bahdanau 等人[10]提出，最先被用于机器翻译领域，并在该领域取得了不错的效果，随后被应用于谷歌神经网络翻译系统。

随着深度学习在情感分析方面的应用，深度学习框架在情感分析方面也得到了越来越多的应用。基于深度学习的文本情感分析的详细讨论可以参见文献[11]。

随着自然语言处理在中文上的发展，常规的基于情感词典和基于机器学习的传统算法在中文文本情感分析上也得到了很大的发展。中文情感分析沿用国外情感分析的方法，结合中文语言的特色，大多数基于有监督学习的研究都取得了不错的成果。深度学习的方法在中文情感分析中也起着重要的作用。文献[12]对中文情感分析的进展做了调研。

2　基于长短期记忆网络和注意力模型的方案

为了实现文本情感分析的目标，本文提出了基于长短期记忆网络和注意力模型的网络结构，模型结构如图 1 所示。该模型主要有两部分组成，一部分是长短期记忆神经网络结构，一部分是注意力模型结构。本文提出的模型的整体流程是：首先对输入的文本句子利用词向量模型来进行编码，将文本转化为词向量表示后，经过基于长短期记忆网络和注意力模型的结构，通过全连接层后通过分类器完成文本情感分析工作。

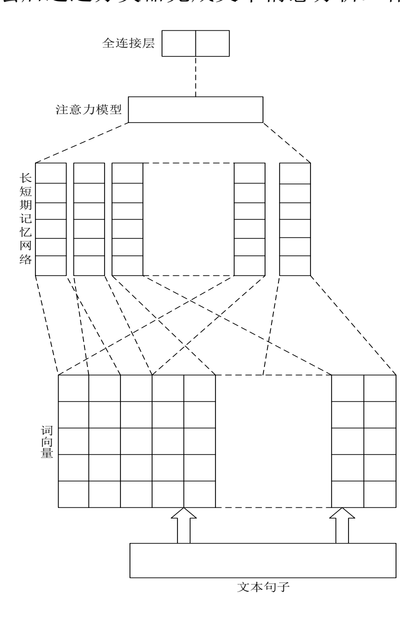


图 1　基于长短期记忆网络和注意力模型的结构图

2.1　模型表示

给定文本句子数据集 D，其中包含有文本 $X\{x_1, x_2, ..., x_m\}$ 和每个句子所对应的情感标签 $Y\{y_1, y_2, ..., y_m\}$，其中每个文本句子 X_i 由 n 个词组成，表示为$\{x_{i1}, x_{i2}, ..., x_{in}\}$，将最终的目标函数表示为

$$p(Y|X) = arg\max_{\theta} f(Y|X;\theta) \tag{1}$$

其中：θ代表该模型中涉及到的所有参数，$f(\cdot)$代表该模型的形式化表达。

2.2　长短记忆网络（LSTM）

循环神经网络的一个核心思想是将以前的信息连接到当前的任务中来，即对以前的信息带有一种记忆的功能。由于文本的上下文关联的特殊性质，因此在文本任务上采用循环神经网络能得到更好的效果。

LSTM 是一种特殊的循环神经网络，是由 Hochreiter 和 Schmidhuber[13]在 1997 年提出的。在许多任务中，LSTM 解决了标准 RNN 的梯度消失和梯度爆炸问题，因此 LSTM 模型表现得比标准的循环神经网络要出色得多。正是因为 LSTM 绝佳的表现，在本文的情感分析任务中，决定采用 LSTM 来进行研究并获取结果。LSTM 的巧妙之处在于通过增加输入门限，遗忘门限和输出门限，使得自循环的权重是变化的，这样使得在模型参数固定的情况下，不同时刻的积分尺度可以动态改变，从而避免了梯度消失或者梯度膨胀的问题。

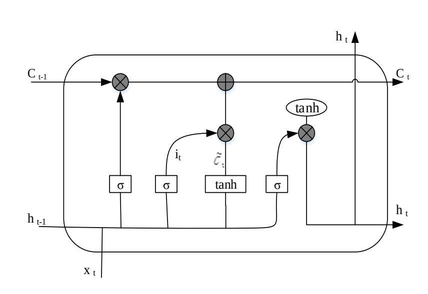


图 2　长短期记忆网络原理图

图 2 是长短期记忆网络（LSTM）的原理图。其中，x 是输入数据，h 为 LSTM 单元的输出，C 则表示记忆单元的值。在 LSTM 动态门结构中，遗忘门决定要忘记什么样的信息，该门在读取 h_{t-1} 和 x_t 之后，输出一个范围在 0 到 1 之间的数值，f_t 则表示要舍弃信息的百分值，0 代表完全舍弃，1 代表完全保留。而 f_t 的计算公式为

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

其中：σ 表示的是 sigmoid 函数，W_f 表示遗忘门权重，b_f 表示遗忘门偏置。

在图 2 中，i_t 指的是更新的值，用于控制当前数据输入对记忆单元状态值的影响。然后，经过 tan$h$ 层将会创建一个新的候选值向量，并被加入到状态中。该过程的两个公式如下：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \quad (3)$$

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

其中：W_i 指的是更新门权重，b_i 表示更新门偏置，tan$h$ 是双曲正切函数，W_c 表示更新候选值，b_c 指的是更新候选值偏置，而 $\widetilde{C_t}$ 指的则是候选值。

接着是更新细胞状态，把旧状态与 f_t 相乘，丢弃掉已经确定的需要丢弃的信息，根据决定更新每个状态的程度进行变化。更新状态的公式如下：

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (5)$$

其中，C_t 指的是新的状态值。

最后，决定输出的值。输出门值 o_t 控制记忆单元状态值的输出，输出信息过程公式如下：

$$o_t = \sigma(W_\sigma \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

其中：W_σ 表示更新输出值的权重，b_o 表示更新输出值偏置，h_t 表示最终确定输出的部分。

2.3　前馈注意力模型

在文本分析任务中，注意力模型是用来表示文本句子中的词与输出结果之间的相关性，该模型最先被应用于机器翻译的任务中。本文采用的前馈注意力模型是对常规的注意力模型的一个直接的简化，该简化方式是从整体序列中构造一个单独的向量 c，构造方式如下：

$$e_t = a(\mathrm{h}_t) \quad (8)$$

$$\alpha_t = \frac{exp(e_t)}{\sum_{k=1}^{T} exp(e_k)} \quad (9)$$

$$c = \sum_{t=1}^{T} \alpha_t h_t \quad (10)$$

其中：a 是一个学习函数，而现在它只由 h_t 决定。在以上的公式中，注意力机制可以被认为是通过计算状态序列 h 的自适应加权平均值来构造输入序列的固定长度的嵌入层 c。

图 3 是前馈注意力模型的示意图，在该图中，隐藏状态序列 h_t 中的向量被馈送到学习函数 a（h_t）中，从而产生概率向量 α。而向量 c 是 h_t 的加权平均值，权值为 α。

注意力机制模型的优势是其随着时间的推移而整合信息的能力。因此，通过使用这种简化的注意力机制，即使 h_t 是采用前馈的方式计算的，模型也可以处理可变长度的序列。由于计算完全可以并行化，因而使用前馈也能导致庞大的效率增益。

本文提出的模型是将长短期记忆模型的输出 h 作为与注意力文本 c 相连接，作为全连接层的输入，最后经过全连接层输出结果。

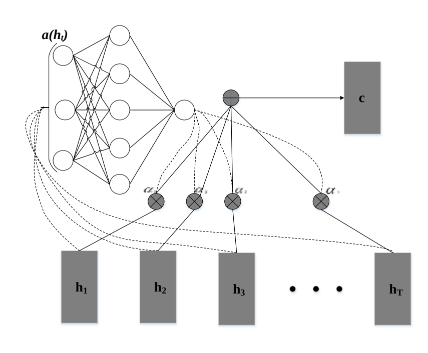


图 3　前馈注意力机制示意图

3　实验与分析

3.1　实验数据准备

为了验证本文提出的模型的有效性，本文选用了有关中文情感挖掘的酒店评论语料（ChnSentiCorp）作为本文提出的模型的测试集，该语料是由中科院谭松波博士收集整理的一个酒店评论的语料，其公布的语料规模为 10000 篇，通过从携程网上自动采集，并经过整理而成。语料被整理成 4 个子集，由于文本情感分析的特殊性，本文采用 ChnSentiCorp-Htl-ba-6000 的数据来进行实验，该语料为平衡语料，包含正负类各 3000 篇。表 1 为数据集样例。

表 1　ChnSentiCorp 数据集样例

积极	消极
地点不错，邻近师范大学，到步行街走路 15 分钟。房间很大很舒适。	没有员工帮客人提行李的服务，房间灯光有点暗，插头小。
店员友善，地点遍历，房间宽敞，住过两次都不错。	肯定我不会再住这里了，太陈旧了，霉味太重，感觉不好！
环境很好，地点很方便，服务也很好，下回还会住。	环境一般，住了之后让人感觉价格和服务不成比例！

3.2　文本预处理

由于中文的特殊性，单独的文字大多数不能独立地表达意

思，需要对中文文本进行分词处理。本文采用结巴分词系统[1]对文本进行分析处理。另外，文本预处理还包括去除停用词等相关步骤，停用词指的是一些没有实际含义的词语。

3.3 建模过程

本文提出的方案建模过程主要采用 Tensorflow 深度学习框架。图 4 是模型框架的构建过程，基于长短期记忆网络和注意力模型的方法进行情感分析的实现方案在前文已经有了介绍。模型构建采用的是 Tensorflow 的序列模型框架，首先添加 Embedding 层作为模型的输入，其次添加 LSTM 模型，并在 LSTM 的模型后添加前馈注意力机制层，在得到句子的向量表示时对评论文本中不同的词赋予不同的权值，然后由这些不同权值的词向量加权得到句子的向量表示。之后添加 Dense 层，以及对 Activation 层采用 sigmoid 函数对文本进行分类。另外，在模型训练过程中，采用 dropout 以防止过拟合。最后，编译过程采用梯度下降算法进行权重的更新迭代。

```
#建立模型
model = Sequential()
model.add(Embedding(len(abc), 256, input_length=maxlen))
model.add(LSTM(128, return_sequences=True))
model.add(Attention_layer())
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

图 4 建模过程部分代码

其中，前馈注意力机制层是按照规定的格式自己构建的。图 5 是前馈注意力机制的构建过程的部分代码，前馈注意力机制层的构建过程严格按照前文介绍的步骤，而代码编写过程也严格按照 Tensorflow 框架要求的格式。

```
def build(self, input_shape):
    assert len(input_shape) == 3

    self.W = self.add_weight((input_shape[-1], input_shape[-1],),
                             initializer=self.init,
                             name='{}_W'.format(self.name),
                             regularizer=self.W_regularizer,
                             constraint=self.W_constraint)
    if self.bias:
        self.b = self.add_weight((input_shape[-1],),
                                 initializer='zero',
                                 name='{}_b'.format(self.name),
                                 regularizer=self.b_regularizer,
                                 constraint=self.b_constraint)

    super(Attention_layer, self).build(input_shape)

def compute_mask(self, input, input_mask=None):
    # do not pass the mask to the next layers
    return None
```

图 5 前馈注意力机制的构建过程

建模完成后，利用本文准备的文本测试数据，将该模型应用于文本情感分析并采用评价指标对模型进行评价。

3.4 实验评价指标

准确率（precision）和召回率（recall）是两个用来评价分类结果好坏的指标，定义如下：

$$\text{precision} = \frac{Correct}{Output} \tag{11}$$

$$\text{recall} = \frac{Correct}{Labeled} \tag{12}$$

其中，Correct 表示系统返回的正确结果数，Output 表示系统返回的结果总数，Labeled 表示测试集中该类的总数。准确率用来衡量分类器的准确性，而召回率则用来衡量分类器是否能找全该类样本，因此这两个指标应该兼顾。用 F_1 测度（准确率和召回率的加权调和平均值）来均衡这两方面，定义如下：

$$\text{F1} = \frac{2*precision*recall}{precision+recall} \tag{13}$$

3.5 实验结果与分析

本文选用维基百科中文语料库作为训练的语料库，利用 Google 开源的 word2vec tool 来进行词向量的训练，然后对语料库进行预处理，并以此作为本文提出的模型的输入层。本文采用的是 word2vec tool 的 skip-gram 模型，上下文窗口大小设置为 5，词向量维度大小设为 5，如果出现有词语不在预先训练好的词向量中，则采用随机初始化的方式来进行表示。

本文对数据集进行了两组实验。第一组是基于不同的文本表示方法对文本进行了分类，确定词向量训练的最佳方案；第二组是基于不同的分类方法对文本进行分类，验证本文提出的情感分析模型的有效性。先来看第一组实验。

BOW-LSTM：将词利用 bag-of-words 模型转化为词向量以后，利用长短期记忆网络进行训练。

if-idf-LSTM：将词利用 if-idf 模型转化为词向量以后，利用长短期记忆网络进行训练。

W2V+LSTM：将词利用 word2vec 训练转换为词向量以后，利用长短期记忆网络进行训练。

表 2 基于不同文本表示方法的分类结果

模型	准确率	召回率	F_1 测度
BOW-LSTM	0.7754	0.7723	0.7738
if-idf-LSTM	0.8423	0.8326	0.8374
W2V-LSTM	0.8643	0.8621	0.8632

对表 2 中的实验结果进行分析：if-idf 模型在 bag-of-words 的基础上对文本中的词进行了加权来表示文本。但是这两种方法都忽略了文本中词语间的语义关系，因此，从模型评价结果来看，W2V-LSTM 得到的结果是三者中表现最好的。由于采用 word2vec 的方法训练词向量后得到的模型效果最佳，因此本文方案也选用 word2vec 的方法对词向量进行训练。

第二组实验是将本文提出的系统与传统的基于 word2vec 的支持向量机（SVM）的机器学习方法，以及基于 word2vec 的卷积神经网络的方法进行对比，利用 ChnSentiCorp-Htl-ba-6000 数据集进行 10 倍交叉验证来进行实验分析，得到的实验结果如表 3 所示。

W2V-SVM：将词利用 word2vec 训练转换为词向量以后，利用 SVM 进行训练分类。

[1] http://oss.io/p/fxsjy/jieba

W2V-Att-CNN：将词利用 word2vec 训练转换为词向量以后，利用基于卷积神经网络和注意力模型的结构进行训练，该模型参照的是冯兴杰等[14]提出的情感分析模型。

W2V- LSTM：将词利用 word2vec 训练转换为词向量以后，利用长短期记忆网络进行训练。

W2V-Att-LSTM：将词利用 word2vec 训练转换为词向量以后，利用基于长短期记忆网络和注意力模型的结构进行训练。

表 3　基于不同分类方法的分类结果

模型	准确率	召回率	F1 测度
W2V-SVM	0.8125	0.8119	0.8121
W2V-Att-CNN	0.8727	0.8713	0.8719
W2V- LSTM	0.8643	0.8621	0.8632
W2V-Att-LSTM	0.8814	0.8798	0.8806

通过实例介绍本文提出的模型的工作步骤，并对表 3 的实验结果进行分析。表 4 中展示了 4 种比较典型的带有情感的文本信息。

首先，在对文本进行预处理后，采用结巴分词系统对文本句子进行分词，分词结果如表 4 所示。分词后采用 word2vec 对词向量进行训练，将训练后的词向量作为模型输入，经过本文提出的 LSTM 模型和前馈注意力模型后的分析，最终，在分类层将输出的文本数据分为积极或消极，通过与已经标注好的数据的比较，可以证明本文模型的有效性。通过对数据样例的分析，很容易看出本文的分析结果在准确性方面是有保障的。

对表 3 中的实验结果进行分析，发现利用 W2V-SVM 运行结果优于 CBOW-SVM 运行结果，这是因为利用 word2vec 训练出的词向量相比于 CBOW 模型而言考虑了上下文的语义信息，因此进行情感分析可以获得更好的结果；对比 W2V-LSTM 的结果优于 W2V-SVM，原因在于利用长短期记忆网络是深度训练模型，得到了更有意义的相关特征并考虑到了上下文之间的关联，从而得到了更好的结果；W2V-Att-LSTM 结果优于 W2V-LSTM，是因为引入的前馈注意力模型机制后考虑到了文本句子和结果的相关性，将前馈注意力模型与长短期记忆模型结合之后，得到了更好的效果。另外，将 W2V-Att-LSTM 与 W2V-Att-CNN 的结果比较，能看出本文长短期记忆网络结合注意力机制的模型比卷积神经网络结合注意力机制的模型表现更佳，更适合用于文本情感分析。

表 4　实例分析展示

文本示例	倾向性	分词结果	模型结果
地点不错，邻近师范大学，到步行街走路 15 分钟。房间很大很舒适。	积极	地点；不错；邻近；师范大学；到；步行街；走路；15 分钟；房间；大；舒适。	积极
肯定我不会再住这里了，太陈旧了，霉味太重，感觉不好！	消极	我；不会；再住；这里；陈旧；霉味；重；感觉；不好。	消极
非常好的酒店，四星的标准完全超值的享受，服务非常好	积极	好；酒店；四星；标准；值；享受；服务；好。	积极
标准间太差，房间还不如三星的，而且设施非常陈旧。建议酒店把老的标准间重新改善。	消极	标准间；差；房间；不如；三星；设施；陈旧；建议；酒店；老；标准间；重新。改善。	消极

4　结束语

本文提出了一种基于长短期记忆网络和注意力模型的文本情感分析方法。实验结果表明了本文提出的方法的有效性和可行性，可以更好地进行文本倾向性的研究，对基于深度学习的文本情感分析有了一定的贡献。

深度学习模型在中文情感分析的发展目前还处于起步阶段，中文语法结构的复杂性、词语意思的多样性，对文本情感分析的发展都有一定的阻碍。对于研究者来说，中文文本情感分析还有很大的发展前景，很值得我们进一步的研究。

参考文献：

[1] Kim S M, Hovy E. Determining the sentiment of opinions [C]// Proc of the 20th International Conference on Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 004: 1367-1374.

[2] Ahmad M, Aftab S, Muhammad S S, *et al.* Tools and techniques for lexicon driven sentiment analysis: a review [J]. Intenational Journal of Multidisciplinary Sciences And Engineering, 2017, 8 (1): 17-23.

[3] Pang Bo, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques [C]// Proc of Conference on Empirical methods in Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2002: 79-86.

[4] Bengio Y, Ducharme R, Vincent P, *et al.* A neural probabilistic language model [J]. Journal of Machine Learning Research, 2003, 3: 1137-1155.

[5] Mikolov T, Sutskever I, Chen Kai, *et al.* Distributed representations of words and phrases and their compositionality [C]// Proc of the 26th International Conference on Neural Information Processing Systems. USA: Curran Associates Inc, 2013: 3111-3119.

[6] Kim Y. Convolutional Neural Networks for Sentence Classification [J]. ArXiv preprint arXiv, 2014: 1408. 5882.

[7] Dos Santos C N, De Bayser M G. Deep convolutional neural networks for sentiment analysis of short texts [C]// Proc of the 25th International Conference on Computational Linguistics. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, 2014: 69-78.

[8] Irsoy O, Cardie C. Opinion mining with deep recurrent neural networks [C]// Proc of Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2014: 720-728.

[9] Tai Kaisheng, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [J]. Computer Science, 2015, 5 (1): 36.

[10] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [J]. ArXiv preprint arXiv, 2014: 1409. 0473.

[11] Zhang Lei, Wang Shuai, Liu Bing. Deep learning for sentiment analysis: A survey [J]. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2018, 8 (7): 1-34.

[12] Peng Haiyun, Cambria E, Hussain A. A review of sentiment analysis research in Chinese language [J]. Cognitive Computation, 2017, 9 (4): 1-13.

[13] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural computation, 1997, 9 (8): 1735-1780.

[14] 冯兴杰，张志伟，史金钏. 基于卷积神经网络和注意力模型的文本情感分析 [J]. 计算机应用研究，2018, 35 (5): 1434-1436. (Feng Xingjie, Zhang Zhiwei, Shi Jinchuan. Text sentiment analysis based on convolutional neural networks and attention model [J]. Application Research of Computers, 2018, 35 (05): 1434-1436.)